INTERNATIONAL JOURNAL OF EMINENT ENGINEERING TECHNOLOGIES

Volume 5, Issue 1 AUG 2017

DESIGN AND IMPLEMENTATION OF NOC ROUTER PERMANENT FAULT TEST ARCHITECTURE

R.V.Naga Supraja[1]
supraja55@gmail.com [1]

V.Siva Sankara Reddy [2]
vssreddy423@gmail.com[2]

[1]pg Scholar, Department of ECE, Krishna Chaitanya Institute of Technology and Sciences,Devaraju gattu, Markapur, Prakasam, Andhra Pradesh.

[2] Assistant Professor ,Department of ECE, Krishna Chaitanya Institute of Technology and Sciences,Devaraju gattu, Markapur, Prakasam, Andhra Pradesh.

ABSTRACT

This project explains the detection of latent hard faults during field operation of Network-on-chip (NoC) by using on-line transparent test technique. Which develop in first input first output buffers. Here, the test algorithm has been integrated in to the router channel interface and on-line test has been performed with synthetic self similar data traffic. This test algorithm involves repeating tests periodically to prevent accumulation of faults. When the test algorithm performance of the NoC after addition of the test circuit has been investigated in terms of throughput while the area overhead has been studied by synthesizing the test hardware. In this, an on-line test technique for the routing logic has been proposed which considers utilizing the headers flits of the data traffic movement in transporting the test patterns.

Index Terms— FIFO buffers, in-field test, NoC, permanent fault, transparent test.

I. INTRODUCTION

Chip integration has reached a stage where a complete system can be placed in a single chip. When we say complete system, we mean all the required ingredients that make up a specialized kind of application on a single silicon substrate. This integration has been made possible because of the rapid developments in the field of VLSI designs. This is primarily used in embedded systems. Thus, in simple terms a SoC can be defined as “an IC, designed by stitching together multiple stand-alone VLSI designs to provide full functionality for an application.” A NoC is perceived as a collection of computational, storage and I/O resources on-chip that are connected with each other via a network of routers or switches instead of being connected with point to point wires. These resources communicate with each other using data packets that are routed through the network in the same manner as is done in traditional networks. It is clear from the definition that we need to employ highly sophisticated and researched methodologies from traditional computer networks and implement them on chip.we have to explore the motivating factors that are compelling the researchers and designers to move toward the adoption of NoC architectures for future SoCs. The area of NoC is still in its infancy, which is one of the reasons why there are various names for the same thing; some call it on-chip networks, some networks on silicon, but the majority agrees upon “Networks on Chips” (NoCs). However, we will be using these terminologies interchangeably

throughout our tutorial.NOC is Integrating various processors and on chip memories into a single chip .Faults occur in NOC

Permanent faults

Transient fault

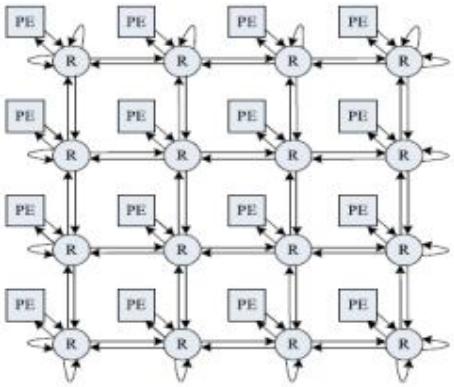


Fig:1. NoC Architecture

II. Literature Survey

As the complexity of designs increases and technology scales down into the deep-submicron domain, the probability of malfunctions and failures in the networks-on-chip (NoCs) components increases. In this work, we focus on the study and evaluation of techniques for increasing reliability and resilience of network interfaces (NIs) within NoC-based multiprocessor system-on-chip architectures. NIs act as interfaces between intellectual property cores and the communication infrastructure; the faulty behavior of one of them could affect, therefore, the overall system. In this work, we propose a functional fault model for the NI components by evaluating their susceptibility to faults. We present a two-level fault-tolerant solution that can be employed for mitigating the effects of both permanent and temporary faults in the NI. Experimental simulations show that with a limited overhead, we can obtain an NI reliability comparable to the one obtainable by implementing the system by using standard triple modular redundancy techniques, while saving up to 48 percent in area, as well as obtaining a significant energy reduction.

"Network-on-a-Chip" (NoC) is a new representation for System-on-Chip (SoC) design. NoC based-systems contain multiple asynchronous clocking that many of today's complex SoC design use. The NoC solution brings a networking method to on-chip communications and claims around a threefold performance increase over usual bus systems. Network-on-Chip (NoC) is a communications essentially composed of routers interconnected by communication channels. It is proper to support the design paradigm. It provides asynchronous communication, scalability, reliability. The idea of network-onchip (NoC) becomes more capable because of performance, power, and scalability requirements for a SoC device. The power consumption in a NoC grows linearly with the amount of bit transitions in subsequent data packets sent through the interconnect design one way to reduce power consumption in NoCs, in both wires and logic, is to reduce the switching activity by means of coding schemes. The interconnect resistance (R) and interconnect capacitance (C) increases linearly with increase in communicate length.

III. PROPOSED ARCHITECTURE

The faults considered in this brief, if applied for SRAMs or DRAMs, can be detected using standard March tests. However, if the same set of faults are

considered for SRAM-type FIFOs, March test cannot be used directly due to the address restriction in SRAM-type FIFOs mentioned in and thus we were motivated to choose single-order address MATS++ test (SOA-MATS++) for the detection of faults considered in this brief. The word oriented SOA-MATS++ test is represented as {_ (wa); ↑ (ra,wb); ↓ (rb,wa)}; _ (ra)} where, a is the data background and b is the complement of the data background. ↑ and ↓ are increasing and decreasing addressing order of memory, respectively. _ means memory addressing can be increasing or decreasing. Application of SOA-MATS++ test to the FIFO involves writing patterns into the FIFO memory and reading them back. As a result, the memory contents are destroyed. However, online memory test techniques require the restoration of the memory contents after test. Thus, researchers have modified the March tests to transparent March test so that tests can be performed without the requirement of external data background and the memory contents can be restored after test. We have thus transformed the SOA-MATS++ test to transparent SOA MATS++ (TSOAMATS++) test that can be applied for online test of FIFO buffers. The transparent SOA-MATS++ test generated is represented as {↑ (rx,w¯ x, r ¯ x,wx, rx)}. The transparent SOA-MATS++ algorithm is intended for test of stuck-at fault, transient fault, and read stuck-at fault, transition fault, and read disturb fault tests developed during field operation of FIFO memories. The fault coverage of the algorithm is shown in Fig. 2. In both the figures, the word size of FIFO memory is assumed to be of 4 bits. As shown in Fig. 2, assume the data word present in lut be 1010.The test cycles begin with the invert phase (memory address pointer j with 0 value) during which the content of location addressed is read into temp and then backed up in the original. The data written back to SOA-MATS++ test.lut is the complement of content of temp. Thus, at the end of the cycle, the data present in temp and original is 1010, while lut contains 0101.Assume a stuck-at-1 fault at the most significant bit (MSB) position of the word stored in lut. Thus, instead of storing 0101, it actually stores 1101 and as a result, the stuck-at-fault at the MSB gets excited.

During the second iteration of j, when lut is readdressed, the data read into temp is 1101. At this point, the data present in temp and\original are compared (bitwise XOR ed). An all 1's pattern is expected as result. Any 0 within the pattern would mean a stuck-at fault at that bit position.

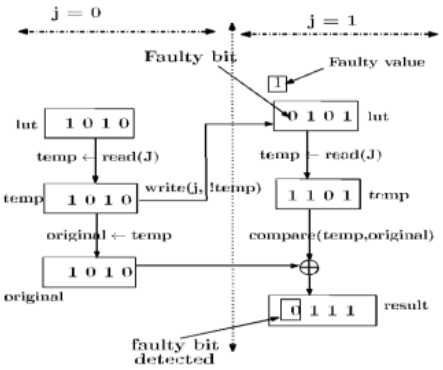


Fig:2. Fault detection during invert phase and restore phase of the transparent SOA-MATS++ test.

where the XOR of 1010 and 1101 yields a 0 at the MSB position of the result indicating a stuck-at-fault at the MSB position. However, for cases where the initial data for a bit position is different from the faulty bit value, the stuck-at-fault cannot be detected for the bit position after the restore phase of

the test. It thus requires one more test cycle to excite such faults.

IV. IMPLEMENTATION OF THE TEST ON FIFO BUFFERS OF NOC ROUTERS

In this section, we present the technique used for implementing the proposed transparent SOA-MATS++ test on a mesh-type NoC. Data packets are divided into flow control units (flits) and are transmitted in pipeline fashion [1]. The flit movement in a mesh-type NoC infrastructure considered for this work is assumed to require buffering only at the input channels of routers. Thus, for a data traffic movement from one core to another, the online test is performed only on the input channel FIFO buffers, which lie along the path. The buffers operate in two modes, the normal mode and the test mode. The normal mode and test mode of operation of a FIFO buffer are synchronized with two different clocks. The clock used for test purpose (referred as test_clk in this brief) is a faster clock compared with the clock required for normal mode (router clock). The FIFO buffers are allowed to be operative in normal mode for sufficient amount of time before initiating their test process. This delay in test initiation provides sufficient time for run-time intermittent faults developed in FIFO buffers to transform into permanent faults. The test process of a targeted FIFO buffer is initiated by a counter, which switches the FIFO buffer from normal mode to test mode. The switching of FIFO buffers from normal mode to test mode occurs after a certain period of time without caring about the present state of the FIFO buffer. It may be argued that at the instant of switching, the buffer may not be full, and as a result not all locations would be tested during the test cycle. However, test initiation after the buffer gets full would cause the following problems. First, wait

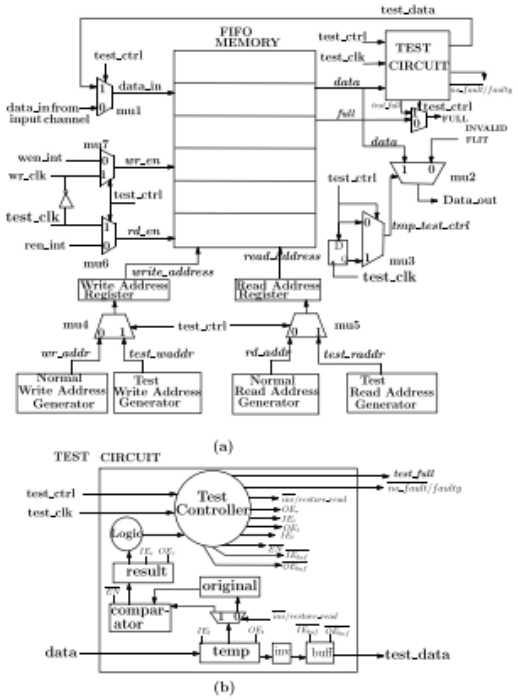


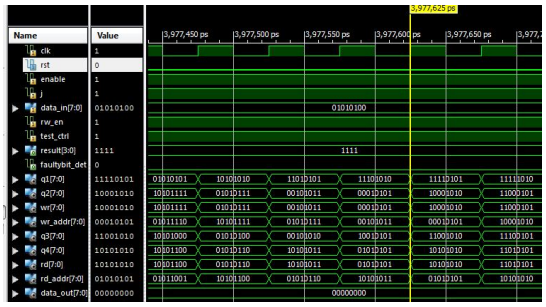
Fig. 3. (a) Hardware implementation of the test process for the FIFO buffers. (b) Implementation of test circuit.

for the buffer to get full would unnecessarily delay the test initiation process and would allow faults to get accumulated. Second, test of the entire buffer would prolong the test time and would negatively affect the normal mode of operation. A test burst involves series of test read and write cycles. It requires three read and two write cycles, or in other words three cycles of the faster test clock to perform a transparent SOA-MATS++ test on a single location of a FIFO buffer. It may be argued that during a test burst, not all FIFO buffer locations are tested or a test of a location can get interrupted. These two problems can be avoided by periodically testing the FIFO buffers. Periodic testing of a FIFO buffer

allows test of a different set of locations of the FIFO buffer in each test burst. Every time the buffer is switched to test mode, the normal process gets interrupted. The FIFO memory location currently addressed in normal mode, at the instant of switching, becomes the target location for test. Since normal operation is interrupted at different instants in different test bursts, the locations tested in each burst would be different. Thus, repeating the test bursts for a number times on a FIFO buffer would cover the test of each location as the number of locations in a FIFO buffer is few. Moreover, periodic testing prevents accumulation of fault in the buffer.

IV RESULTS

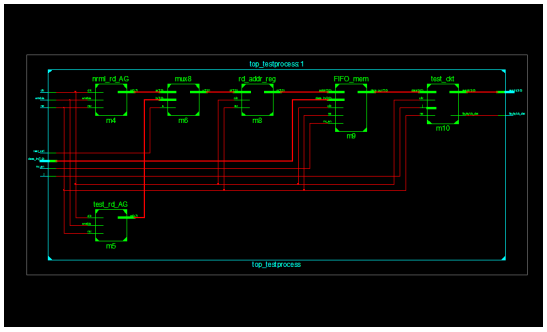
Simulation



Area

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	265	4656	5%
Number of Slice Flip Flops	291	9312	3%
Number of 4 input LUTs	434	9312	4%
Number of bonded IOBs	12	232	5%
Number of GCLKs	2	24	8%

RTL Schematic



V. CONCLUSION

This project detects the run time permanent faults developed in SRAM-Based FIFO buffer by using SOA-MATS++ test algorithm. The test algorithm is used to perform on-line and periodic test of FIFO buffer which is present within the routers of the NoC. The test algorithm performances based on the look up tables. Here testing of buffers presents accumulation of faults and also allows test of each location of the buffer. The simulation result shows that periodic testing of FIFO buffers do not have much effect on the overall throughput of the NoC except while buffers are tested too frequently.simultaneosuly the proposed on-line test technique for the routing logic with the test buffers involves utilization of the unused fields of the incoming data packets for the pattern encoding.

REFERENCES

[1] W. J. Dally and B. Towles, “Route packets, not wires: On-chip interconnection networks,” in Proc. 38th Annu. Design

Autom. Conf., 2001, pp. 684–689. [2] A. Bondavalli, S. Chiaradonna, F. Di Giandomenico, and F. Grandoni, "Threshold-based mechanisms to discriminate transient from intermittent faults," IEEE Trans. Comput., vol. 49, no. 3, pp. 230–245, Mar. 2000. [3] M. Radetzki, C. Feng, X. Zhao, and A. Jantsch, "Methods for fault tolerance in networks-on-chip," ACM Comput. Surv., vol. 46, no. 1, pp. 1–38, Jul. 2013, Art. ID 8. [4] S. Ghosh and K. Roy, "Parameter variation tolerance and error resiliency: New design paradigm for the nanoscale era," Proc. IEEE, vol. 98, no. 10, pp. 1718–1751, Oct. 2010. [5] S. Borri, M. Hage-Hassan, L. Dilillo, P. Girard, S. Pravossoudovitch, and A. Virazel, "Analysis of dynamic faults in embedded-SRAMs: Implications for memory test," J. Electron. Test., vol. 21, no. 2, pp. 169–179, Apr. 2005. [6] M. Bushnell and V. Agrawal, Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits (Frontiers in Electronic Testing). New York, NY, USA: Springer-Verlag, 2000. [7] D. Xiang and Y. Zhang, "Cost-effective power-aware core testing in NoCs based on a new unicast-based multicast scheme," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 30, no. 1, pp. 135–147, Jan. 2011. [8] K. Petersen and J. Oberg, "Toward a scalable test methodology for 2D-mesh network-on-chips," in Proc. Design, Autom., Test Eur. Conf. Exhibit., Apr. 2007, pp. 1–6. [9] D. Xiang, "A cost-effective scheme for network-on-chip router and interconnect testing," in Proc. 22nd Asian Test Symp. (ATS), Nov. 2013, pp. 207–212. [10] M. Ebrahimi, M. Daneshtalab, J. Plosila, and H. Tenhunen, "Minimal-path fault-tolerant approach using connection-retaining structure in networks-on-chip," in Proc. 7th IEEE/ACM Int. Symp. Netw. Chip (NoCS), Apr. 2013, pp. 1–8. [11] C. Grecu, P. Pande, B. Wang, A. Ivanov, and R. Saleh, "Methodologies and algorithms for testing switch-based NoC interconnects," in Proc. 20th IEEE Int. Symp. Defect Fault Tolerance VLSI Syst., Oct. 2005, pp. 238–246. [12] M. R. Kakoee, V. Bertacco, and L. Benini, "A distributed and topologyagnostic approach for on-line NoC testing," in Proc. 5th ACM/IEEE Int. Symp. Netw. Chip, May 2011, pp. 113–120. [13] S. Barbagallo et al., "A parametric design of a built-in self-test FIFO embedded memory," in Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst., Nov. 1996, pp. 221–229. [14] A. J. van de Goor and Y. Zorian, "Functional tests for arbitration SRAM-type FIFOs," in Proc. 1st Asian Test Symp. (ATS), Nov. 1992, pp. 96–101. [15] M. Nicolaidis, "Theory of transparent BIST for RAMs," IEEE Trans. Comput., vol. 45, no. 10, pp. 1141–1156, Oct. 1996. [16] S. Kundu, J. Soumya, and S. Chattopadhyay, "Design and evaluation of mesh-of-tree based network-on-chip using virtual channel router," Microprocess. Microsyst., vol. 36, no. 6, pp. 471–488,